October 1994
Volume 76. No. 10
Pages 727–802
$4.00 US

JOURNAL of the PATENT and TRADEMARK OFFICE SOCIETY

In This Issue

- The *Alappat* Standard for Determining That Programmed Computers are Patentable Subject Matter

The *Alappat* Standard for Determining that Programmed Computers are Patentable Subject Matter

James R. Goodman[1], *Todd E. Marlette*[2], *and Peter K. Trzyna*[3]

The view of *In re Alappat*[4] that "programming creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed...." is the key to understanding how patentable subject matter requirements apply to computer program-related inventions. This Article[5] examines this view and its significance to understanding why a programmed computer must be treated as a "machine" pursuant to 35 U.S.C. § 101.

I. Overview

Under existing statutes, a programmed computer should or should not be protectable through the patent system based on the following general principles: (A) the statutes governing patentable subject matter[6]

---

1 James R. Goodman is a Professor of Computer Science, University of Wisconsin, Madison.

2 Todd E. Marlette is an associate at Keck, Mahin & Cate, Washington, D.C.

3 Peter K. Trzyna is a partner at Keck, Mahin & Cate, Chicago, Illinois.

4 31 USPQ2d 1545, Case No. 92-1381, 1994 WL 395 740 (Fed. Cir. July 29, 1994).

5 This article is based on a filing by Peter K. Trzyna in the matter of the Notice of Public Hearings and Request for Comments on Patent Protection for Software-related Inventions in the Federal Register, Vol. 58, No. 242, Dec. 20, 1993.

This article reflects only the personal views of the authors and is not offered to represent the views of any institution, client, applicant for a patent, or other person.

Thankful appreciation is expressed to the following individuals who helped guide the thinking herein: Lee Hollaar, Professor of Computer Science, University of Utah, Salt Lake City; Alexander C. Johnson, Jr., Marger, Johnson, McCollom & Stolowitz, P.C., Portland; James A. Sprowl, Fitch, Even, Tabin & Flannery, Chicago; William D. Thomson, Thomson Patent Services, Arlington (Virginia); and Michael R. Williams, Professor of Computer Science, University of Calgary, Calgary (Canada).

6 35 U.S.C. §§ 100, 101.

apply uniformly to all machines and processes for making and using them; there is no statutory basis for treating programmed computers different from other machines; (B) a computer program literally makes a new machine; (C) if an unprogrammed computer is viewed as a known machine, an operating computer program is a process involving a use of a known machine; and (D) if the patent law does not recognize that hardware and computer program embodiments are equivalent circuits, then a patent on either one can be evaded—often easily—by using the other.

A. *Uniform Application of the Law*

In determining what constitutes patentable subject matter, there is no statutory basis for treating computer program-related inventions differently from other kinds of inventions. What constitutes a "machine" was determined by Congress in enacting 35 U.S.C. § 101, which provides that

> Whoever invents or discovers any new and useful process, machine... or any new and useful improvement thereof, may obtain a patent therefor....

Also, 35 U.S.C. § 100 defines a "process" a including "a new use of a known process, machine...."

The term "machine" in this and forerunner statutes has been defined as an instrument; that is to say a physical entity, consisting of parts, components, or elements, which are so arranged and organized as to cooperate, when set in motion, to produce a definite, predetermined, and unitary result.[7] "The term 'machine'... embraces not only machines as commonly understood, but various mechanisms, mechanical elements and combinations."[8] The particular way in which the components are arranged, as well as the nature of the components themselves, are the parameters which define and distinguish a machine.[9] And it has long been recognized that the patentability of a machine may reside entirely in the manner in which the components are arranged, even if all of the components are old.[10]

The Board of Patent Appeals and Interferences has recognized that the material or workpiece upon which a machine is capable of operating is not considered for determining patentability of the machine.[11] Ap-

7 *Burr v. Duryee*, 68 U.S. (1 Wall.) 531, 570–71 (1863).

8 1 Chisum, *Patents* §1.02[1] (Aug. 1993) *quoting* 1 A. Deller, *Walker on Patents* §119 (2d ed. 1964).

9 Rosenberg, *Patent Law Basics*, §6.01[2] [a], Machine (Clark, Boardman, Callaghan 1993).

10 *Seymour v. Osborne*, 78 U.S. (11 Wall.) 516 (1870).

11 *Ex Parte Masham*, 2 U.S.P.Q.2d 1647, 1648 (P.T.O. Bd. Pat. App. & Int. 1987).

plying this approach consistently to a programmed computer, the "material or workpiece" could be viewed as the known electrical charge which moves through the computer circuitry. In an unrelated example, the Board has emphasized that it is clearly not obvious to combine pieces of prior art where structures disclosed therein are used for different purposes.[12] Therefore, when a computer program operates to control a computer, although the actual transistors in the computer are known and used for many different purposes, a particular combination and connection of transistors can meet the requirements of patentability.

B. *A Computer Program Literally Makes a New Machine*

1. *Early Computers Were Manually Programmed to Make New Machines*

The first general purpose electronic computer was the ENIAC, a portion of which is on exhibit at the Museum of American History in Washington, D.C. As can be seen from the exhibit, programming the ENIAC was done manually with telephone patch cords inserted into plugboards. The patch cords would connect calculating units, which were further programmed by setting mechanical switches on the calculating units.[13] Quite literally, the attaching of the cables made new circuitry, *i.e.*, a special purpose computer. Thus, *the only difference* between a programmed general purpose computer (plugged in patch cords and set switches) and a hardwired circuit is solder.

Using wiring and mechanical switches was not a new way to program or customize a general purpose machine. The same technique can be found in ENIAC forerunners such as the Colossus machine used to decipher coded German communications during World War II.[14]

Later computers also used mechanical programming, but sought ways to store the programs. The Mark I computer, for example, was programmed by using a 32 pushbutton manual keyboard.[15] The computer program was then stored with vacuum tubes.

---

12 *Ex parte Kice*, 211 U.S.P.Q. 560, 562 (P.O. Bd. App. 1980).

13 See A. Burks, "The ENIAC: First General Purpose Computer," *Annals of the History of Computing*, Vol. 3, No. 4, pp. 310–399, (Oct. 1981) and C. Bashe et al., *IBM's Early Computers*, (MIT Press 1986) (in particular Chapter 1, Punched Cards and Plugwires). See also P. Ceruzzi, "An Unforseen Revolution: Computers and Expectations," *Imagining Tomorrow* (J. Corn, ed., MIT Press 1986), which shows a photograph on p. 187 of the ENIAC programming. "This rewiring essentially changed it into a new machine for each new problem it solved." P. Ceruzzi at p. 196.

14 See, for example, T. Flowers, "The Design of Colossus," *Annals of the History of Computing*, Vol. 5, No. 3, pp. 239–259 (July 1983) (note particularly the programming switches in Figure 3). See also, W. Chandler, "The Installation and Maintenance of Colossus," *Annals of the History of Computing*, Vol. 5, No. 3, pp. 260–262 (July 1983).

15 See, for example, M. Campbell-Kelly, "Programming the Mark I," *Annals of the History of*

Early circuitry such as that used in relay computers, vacuum tube computers, and parametron computers (no tubes or transistors, but circuitry equivalent thereto) was subsequently replaced with solid state circuitry.[16] Solid state circuitry has since been reduced in scale with advances in semiconductor technology. Although today's circuitry is smaller, a computer program still performs in the same manner, *i.e.*, it creates new circuitry configurations through its operation.

2. Macroscopic View of the Physical Structure of a Modern General Purpose Computer

A macroscopic view of the physical structure of a general purpose computer shows some or all of the following components: Random Access Memory (RAM), Read Only Memory (ROM), Cathode Ray Tube (CRT), Central Processing Unit (CPU), Disk Storage Unit, Access Ports, Printer, Modem, Keyboard and Mouse.[17] A computer program determines the physical connection and interaction between the above-identified components.

On this macroscopic level, a different machine is created each time the various components of a computer are connected. A computer having a CRT, CPU, and keyboard is quite different from one having only a CRT, CPU, and Mouse. Each of the above combinations cooperates differently to produce a definite, predetermined, and unitary result, and is therefore a separate and distinct machine. This is so even if each of the components remain at the same work station, connected by patch cords, while a computer program electrically connects and disconnects the various components.

Most would agree that today, a combination of the above-identified components would be anticipated or obvious under 35 U.S.C. §§ 102,

Computing, Vol. 2, No. 2, pp. 130–168 (Jan. 1980); K. Zuse, "Installation of the German Computer Z4 in Zurich in 1950," *Annals of the History of Computing*, Vol. 2, No. 2, pp. 239–245 (Jan. 1980) (discussing the earlier use of punched tape to store programs); C. Hurd, "Early IBM Computers: Edited Testimony," *Annals of the History of Computing*, Vol. 3, No. 2, pp. 163–182 (Jan. 1981) (discussing replacement of mechanical storage devices with tubes). Note that the early use of programmed computers to perform certain financial and business calculations is also described therein. A first program assembled financial data from departments of Monsanto Chemical Corporation and prepared quarterly reports, while a second program handled payroll. The development of Random Access Memory (RAM) is mentioned in C. Evans, "Conversation: Jay W. Forrester," *Annals of the History of Computing*, Vol. 5, No. 3, pp. 297–301 (Jan. 1983). See also C. Hurd, "Early IBM Computers: Edited Testimony," *Annals of the History of Computing*, Vol. 3, No. 2, pp. 163–182 (Jan. 1983).

16 See, for example, H. Takahasi, "Some Important Computers of Japanese Design," *Annals of the History of Computing*, Vol. 2, No. 4, pp. 330–337 (Jan. 1980).

17 See generally, Harman et al., *The Motorola MC 6800 Microprocessor Family*, Chapter 1.1, Application of the Motorola MC68000 (Prentice-Hall 1985).

103. But that does not mean that every process by which these components are used is obvious or anticipated. A new use of old patentable subject matter can be patented.[18] More importantly, the question of novelty is separate from the statutory subject matter question under 35 U.S.C. § 101.[19] However, even on the macroscopic level, combinations of components may be neither anticipated nor rendered obvious. One could easily envision a complicated manufacturing system comprising a plurality of specialized electrical components, such as electric motors, for operating a plurality of hydraulic presses. Each component would be switched into and out of operation through a corresponding plurality of electrical connections. Such a system would clearly form a plurality of new machines, apart from the unused components, as each electrical component is utilized. Each individual manufacturing machine, if novel and nonobvious, would therefore be entitled to patent protection.

3. Microscopic View of the Physical Structure of a Modern General Purpose Computer

While a computer program may work at the macroscopic level to control complex machinery, it also works on the microscopic, or sub-architectural level. From a microscopic view, most modern digital computers are comprised of Large Scale Integrated (LSI) and Very Large Scale Integrated (VLSI) electronic circuits.[20] A digital computer program interacts with the electric circuitry to form a multitude of instructions which are stored for execution according to a predetermined protocol.[21] Each computer program thus interacts with hardware such that parts and components are connected to produce a definite, predetermined, and unitary result, *i.e.*, a machine.[22]

Each computer instruction, when executed, forms a unique physical pathway within the computer. While the execution of a computer program instruction can be quite complicated, it can be illustrated with a simple NAND gate circuit.

In response to a received pair of electrical signals, a NAND gate outputs a predetermined signal having a value of "0" representing 0

18 The test for patentability of a method directed to a new use of an old compound is the unobviousness of the new use, not of the compound. *In re Sebek*, 347 F.2d 632 (C.C.P.A. 1965); if the result of the process is unobvious and the particular use of the material is not suggested by the prior art, the process is patentable. *Ex parte Wagner*, 88 U.S.P.Q. 217 (P.O.Bd.App. 1950).

19 *Diamond v. Diehr*, 450 U.S. 175 (1981).

20 Pucknell et al., *Basic VLSI Design Systems and Circuits*, Preface (Prentice-Hall 1988).

21 See generally, Harman et al., *The Motorola MC 6800 Microprocessor Family*, Chapter 5, MC68000 Assembly Language and Basic Instructions (Prentice-Hall 1985).

22 *Id.*

volts or a value of "1" representing +5 volts.[23] For each of two input signals A and B, the NAND gate output X corresponds to the following table:

A	B	X
0	0	1
1	0	1
0	1	1
1	1	0

By manipulating a plurality of transistors within the NAND gate, a unique physical pathway is created for each of the four electronic signal combinations. Each of a plurality of transistors contained within the NAND gate electronically forms either a closed circuit or an open circuit in response to the input. Thus, the NAND gate pathway is modified for each combination of variables A and B to create four distinct machines.

While data storage may be accomplished in any number of devices ranging from an abacus to a classical neuron, modern computer programs are generally executed on, and stored by, computers having many transistors.[24] Transistors, as used in modern general purpose computers, are created on bare wafers of polished silicon or other semiconductor material.[25] The wafer surface is exposed to fabrication techniques such as the application of photo-resist patterns, high temperature oxidation, and deposition and etching processes,[26] to form transistors.[27]

A semiconductor transistor (created on the silicon wafer) physically responds to electrical signals applied to its gate, source, and drain. For example,[28] an nMOS enhancement mode transistor is formed from a section of silicon wafer known as p-substrate silicon, *i.e.*, neutral silicon that has been doped with an impurity such as Boron to create

23 Pucknell et al., *Basic VLSI Design Systems and Circuits*, Chapter 5.3.2, Two-input NAND Gate, p. 102 (Prentice-Hall 1988).

24 See generally, Mead, *Analog VLSI and Neural Systems* (Addison-Wesley 1989). More specifically, Chapter 4, Neurons, Figure 4.1, p. 44 shows a detailed view of the neuron as an information storage cell.

25 Mead, *Analog VLSI and Neural Systems*, Appendix A, Patterning, pp. 306-308 (Addison-Wesley 1989).

26 Examples include chemical vapor deposition (CVD), sputter deposition, local oxidation (LOCOS) and plating.

27 *Id.* See also, Mead et al., *Introduction to VLSI Systems*, Chapter 1.7, nMOS fabrication, p. 10 (Addison-Wesley 1980).

28 Only a very cursory explanation of the nMOS enhancement mode transistor is given for purposes of illustration.

an excess of mobile positive charges.[29] The p-substrate includes therein a pair of n-diffusion regions or electron rich regions created through the deposition of impurities such as phosphorous. The n-diffusion regions are naturally separated from the p-substrate by a pair of depletion regions. A pair of electrodes are connected to each of the n-diffusion regions and a gate electrode is disposed therebetween. If no voltage is applied to the gate electrode, no electrical channel is formed between the drain and source and the transistor does not conduct electricity. The circuit is electrically an open circuit.[30]

The physical structure of the nMOS enhancement mode transistor changes as it is placed into conductive operation.[31] Conductive operation is achieved by placing a voltage potential across the gate and source (V_{GS}) which is greater than a threshold voltage VT. With specific reference to the channel, an "inversion region" is physically formed which connects the n-type source to the n-type drain. Thus, through the application of voltage (V_{GS}), the p-substrate region is transformed into a region of excess electrons, thereby creating an electrical pathway for electricity. The circuit is electrically a closed circuit.[32] Thus, the physical presence or absence of the electrical channel forms a pair of unique physical circuits, *i.e.*, a closed or open circuit, respectively.

i. Memory Cells Are Configured by Storing A Computer Program

A computer program is commonly stored within the memory cells of a digital system.[33] Each memory cell is used to store one unit or "bit" of data. The memory cells are commonly combined with other memory cells to form Random Access Memories or RAMs.

To better understand the physical structure of a Random Access Memory, a single memory cell is examined. In an illustrative example,[34] a J-K flip flop may be formed from a plurality of NAND gates (as

29 As representatively shown by Figure 1-4(a) of Mead et al., *Introduction to VLSI Systems*, Chapter 1.4, Basic nMOS Transistors, Figure 1-4(a), p. 7 (Addison-Wesley 1980).

30 Absent, of course, the presence of a breakdown voltage between the source and drain. However, even air itself may be a conductor given sufficient voltage, *e.g.* lightning.

31 See Figure 1-5(a) of Mead et al., *Introduction to VLSI Systems*, Chapter 1.4, Basic nMOS Transistors, Figure 1-5(a) p. 8 (Addison-Wesley 1980).

32 This is, of course, for values of the effective gate-source voltage (gate-source voltage minus the threshold voltage) which are much greater than the drain-source voltage, *i.e.*, V_{GS} - VT >> V_{DS}.

33 Bakoglu, *Circuits, Interconnections and Packaging for VLSI*, Chapter 4.3, Random Access Memory (RAM), p. 143 (Addison-Wesley 1990).

34 An example of a memory cell is shown in Figure 10-12(a) of Pucknell et al., *Basic VLSI Design Systems and Circuits*, Chapter 10, Memory, Registers, and Aspects of System Timing, p. 220 (Prentice-Hall 1988).

previously discussed).[35] A J-K flip flop has essentially two input lines, J and K, an output Q and a clock input or trigger. For each combination of predetermined input values along input lines J and K at the time that the clock signal is triggered, a predetermined electrical pathway is formed which corresponds to the values of each of the NAND gates contained therein. Thus, just as each of the individual NAND gates forms a distinct machine in response to predetermined input, a combination of NAND gates, in the form of a J-K flip-flop, will form a distinct machine in response to its input.

A second kind of memory cell is the CMOS static memory cell which may be formed directly from six (6) separate transistors.[36] Each of the transistors may be characterized as an open circuit or as a short circuit in response to the input data lines or "BIT" lines. Thus, the same way that a NAND gate forms a new circuit based upon the A and B inputs, the CMOS static memory cell forms a new circuit based upon the "BIT" line inputs.

Most memory cells, including J-K flip flops and CMOS static memory cells, incorporate a pair of cross-coupled, self-driving circuits.[37] Each circuit remains indefinitely in either of two distinct states to retain information.[38] Each RAM cell, through its transistors, forms one of two distinct and separate circuits.

Memory cells are commonly manufactured in very large numbers. Today's technology permits the creation of dynamic RAMs with an excess of 16 Mbit (16 million memory cell) capacity.[39] With RAMs being formed in such large numbers, and with each memory cell forming at least one of two predetermined electrical pathways, the number of separate machines from this component alone may be on the order of $2^{16,000,000}$ (*i.e.*, 1 followed by more than 5 million zeros). Combinations of transistors may thus be formed which satisfy the requirements of novelty and non-obviousness as prescribed by Title 35, United States Code, §§ 102, 103.

---

35 The J-K flip flop as shown in Pucknell, et al. is constructed from 9 NAND gates.

36 See Figure 4.7(a) of Bakoglu, *supra*, at 144.

37 Accordingly, *In re Edward S. Lowrey*, ____ F.2d ____, (Fed. Cir. August 26, 1994), is correct as a matter of fact that a configuration of memory cells is patentable subject matter (as a new machine); the Patent and Trademark Office view, to the contrary, that it is printed matter, is scientifically untenable.

38 Mead et al., *Introduction to VLSI Systems*, Chapter 1.14, Properties of Cross-Coupled Circuits, p. 26 (Addison-Wesley 1980.).

39 As shown by Bakoglu in Table 4.1. See also, Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Chapter 4.3, Random Access Memory (RAM), p. 145 (Addison-Wesley 1990).

ii. Interaction Between A Computer Program Instruction and a Computer

To understand the interaction between a computer program instruction and a computer, consider that memory cell arrays may have the capability to form many different machines. However, a protocol is needed to provide useful electrical connections. Groups of individual memory cells may be combined to store a command or an "assembly language instruction."[40] A typical assembly language instruction may be the size of a byte (8 adjacent memory cells or "bits"), a word (16 "bits"), or a long word (32 "bits"). Each assembly language instruction may then be combined into a listing for execution on a computer.[41] As instructions are loaded into the computer memory, the corresponding machine is created. While individual assembly language instructions may have lost their novelty through public use by a computer, combinations of instructions may be created which satisfy the requirements of patentability.

Each assembly language instruction is designed to complete a specific task within a microprocessor. In executing a computer program, it makes no difference whether a hard wired (*i.e.*, special purpose) computer is used instead of a programmed computer. Two machines can be considered equivalent as a matter of computer science if for every possible input sequence, the same output sequence is generated.

Because there are numerous different microprocessors, each having a unique language, it is generally inefficient for programmers to work directly with assembly language. Higher level programming languages such as Fortran, Pascal, Cobol, and "C" provide a more accessible format for humans. Translation between higher level programming languages and a language understood by the computer hardware is generally performed by a computer program known as a compiler.

A computer program that has been translated into the language understood by the hardware may be stored in a volatile memory such as a RAM, a nonvolatile memory such as PROM, FRAM, or battery backed RAM, or hardwired into a permanent storage such as a ROM (Read Only Memory). A ROM may take various forms including: punch cards (such as those used with the first IBM mainframe computers); locked floppy disks or other magnetic media; integrated circuits (using a plurality of fuses, antifuses, or transistors); EPROMs; or optical storage media (such as the bar codes found on grocery items and com-

40 See Table D-12, Harman et al., *The Motorola MC 6800 Microprocessor Family*, Chapter 12, Miscellaneous Instruction Times, p. 533 (Prentice-Hall 1985).

41 *Id.* Chapter 5.1, Software Development, p. 124.

pact discs). ROM instructions are functionally equivalent to the instructions stored in a RAM. In fact, ROM instructions are routinely loaded into a RAM for execution by a computer.

iii. The Means for Storing A Computer Program Has No Bearing on Its Operation

Diamond v. Bradley 450 U.S. 381 (1981) can be interpreted as establishing that ROM-implemented processes are patentable even if the same process, computer program implemented, would not. However, at least one legal commentator, D. Davidson,[42] observed the following:

> This would be silly. The ROM in *Bradley* could have been replaced in the computer with a volatile RAM (read-write) memory chip which looks like a ROM; the software could first be read into the RAM, then the process performed.[43]

The *Bradley* decision is indefensible from a factual *i.e.*, scientific, point of view.[44]

A computer program, however stored, operates in a computer to set switches, to make electrical connections, and to form circuitry. The exact same circuitry or equivalent circuitry can be hardwired so that the connections are more permanent. In any case, there is a new machine.

This is, in part, where the dissenting opinion of *In re Alappat* loses the thread. The dissent draws an analogy from storing music on a piano player roll or a compact disc to a program operating on a computer.[45] It is a false analogy. More correct is to draw an analogy between a compact disc and a floppy disc, as both can be used to store a program. No one contends that a new piece of music makes a new piano, but programming a computer, as shown above, literally makes a new, *i.e.*, distinct or different, configuration of the circuitry. The dissent also states the following at page 92:

> Yet Alappat also concedes that a claim drawn to "a method which amounted to a mathematical algorithm [without] any disclosed hardware or structure, other than a programmed general purpose computer," is nonstatutory. Br. for Alappat at 22.

42 D. Davidson, "Protecting Computer Software," *Arizona State Law Journal* Vol. 1983, No. 4, pp. 612–798.

43 *Id.* at 646.

44 Compare, *In re Iwahashi*, 888 F.2d 1370 (Fed. Cir. 1989); see footnote 24 in *Alappat.*

45 But see P. Ceruzzi at ftnt. 50, *infra.*

Based on this purported quote, the dissent argues that the invention cannot be viewed as patentable as a circuit and unpatentable as a programmed computer. Actually, Judges Archer and Nies are mistaken. Alappat's brief makes no such statement.[46] Alappat instead viewed an appropriately programmed computer as an equivalent to the circuit.[47]

Additionally, the dissenting opinion states the following at footnote 26:

> ... improved digital circuitry itself, such as faster digital processors, would be statutory subject matter. Unlike the "rasterizer" in this case, they are not simply a claimed arrangement of circuit elements defined by a mathematical operation which does nothing more than solve the operation that defines it....

The dissenting opinion does not recognize that, as a matter of fact, the "rasterizer" is a form of "improved digital circuitry." This is true regardless of whether the circuitry is soldered together or connected by programming a computer. In addition, it is rather common in electrical engineering to define digital circuitry functionally by the mathematical operation it performs.[48] As stated by Judge Rader "Mathematics is simply a form of expression—a language."[49]

4. Summary

When a computer is programmed, a new machine is created as a matter of fact. Programming a computer creates a new machine because a general purpose programmable computer becomes a special purpose computer once it is programmed to perform particular functions. The computer program connects many different parts of the computer in a new or different configuration. If these parts are connected in a way that is new and useful, patent protection is required under 35 U.S.C. § 101.

46 The dissenting opinion misinterprets a summary of the law applied by the Board as an admission of *Alappat*. Page 22 of the Alappat brief merely commented on the treatment by courts of apparatus and method claims in the *Maucorps, Walter, Meyer, Pardo,* and *Abele* cases. (Citations omitted)

47 This position was taken by Alexander C. Johnson, Jr. during the oral argument for the appeal.

48 See, for example, Fredrick Hill and Gerald Peterson, *Introduction to Switching Theory & Logical Design*, 3d, (John Wiley & Sons, 1981) (see particularly chapter 15 and 16).

49 *Alappat,* 31 USPQ2d 1545, Case No. 92-1381, *1994 WL 395 740 (Fed. Cir. July 29, 1994)* (concurring opinion of Judge Rader at ¶ 8).

C. *If an Unprogrammed Computer Is Viewed as a Known Machine, an Operating Computer Program is a Process Involving a Use of a Known Machine*

As stated above, a computer program makes a new machine structure. Specifically, "...the computer itself was not just a 'machine,' but at any moment it was one of an almost infinite number of machines depending on what its program told it to do."[50] P. Ceruzzi made the following observation:

> The computer, by virtue of its programmability, is not a machine like a printing press or a player piano—devices which are configured to perform a specific function. By the classical definition, a machine is a set of devices configured to perform a specific function; one employs motors, levers, gears, and wire to print newspapers; another uses motors, levers, gears, and wire to play a prerecorded song. A computer is also made by configuring a set of devices, but its function is not implied by that configuration. It acquires its function only when someone programs it. Before that time it is an abstract machine, one that can do "anything." (It can even be made to print a newspaper or play a tune.) To many people accustomed to the machines of the Industrial Revolution, a machine having such general capabilities seemed absurd, like a toaster that could also sew buttons on a shirt. But the computer was just such a device; it could do many things its designers never anticipated.[51]

Just because it is *adaptable*, does not make it any less a machine. Accordingly, even if the unprogrammed computer is viewed as a known machine, setting the switches to make new circuits by means of the computer program is a process or method involving "a new use of a known...machine" which is literally all that is required for patentable subject matter under 35 U.S.C. §§ 100, 101.

D. *Hardware and Software Embodiments Must be Equivalent to Prohibit Duplication in Each Respective Medium*

If a hardwired computer and a programmed computer are not recognized as equivalents, then obviously a patent on one can be avoided by using the other.

1. Sample U.S. Patents

Consider such patents as U.S. Patent Nos. 3,597,742 and 4,135,241, the latter of which is a continuation-in-part of the former

50 P. Ceruzzi in "An Unforseen Revolution: Computers and Expectations, 1935–1985," *Imaginging Tomorrow* at p. 196.
51 *Id.* at pp. 196–197.

and has the following as an abstract, which seems to discuss a computer program-related invention:

A data handling system for a hospital or like establishment. The system keeps track of bed allocation, changes in inventory, and charges to patients, and also serves as a communication network for the hospital. Data is fed into the system in the form of pre-punched cards bearing patient information, inventory data, and commands or messages, and thus unskilled personnel can quickly feed data into the system without error. Message data is routed directly to teleprinters at addressed locations. Bed allocation and patient data, and charge and inventory data are respectively stored in separate magnetic drum storage areas. Searching facilities are provided which can locate desired data entries in either storage area and mark these entries for printout, and separate printout circuitry then transfers marked data items to the proper addresses in the proper format. At the end of each day, a final search is performed which produces a printout of all charges organized by patient number. A tally inventory search is also performed which produces a printout of inventory changes organized by item number and by department number. The tally search is cumulative, and a tally arithmetic unit summarizes inventory data for each separate item before printout. A running record is kept of each day's total charges, credits, and payments on account in a central core memory, and this record is continually updated by a central arithmetic unit.

While is may seem from the abstract that the patents are directed to a data processing system implemented by a programmed general purpose computer, in fact, the patents cover pure hardwired logic. Of course it would be relatively easy to do the same with a general purpose computer because the program would make equivalent circuitry. Without patent law covering both embodiments, it would be easy to design around the hardware embodiment limitations implicit in the "means-for" claims.[52]

2. Sample Court Decisions

Consider the following court opinions. The Court in *The Magnavox Co. and Sanders Assoc., Inc. v. Mattel, Inc.*[53] found infringement of U.S. Patent No. Re. 28,507, involving analog circuitry by a programmed computer system and held that "The Rusch display circuit... and the Mattel display processor are two techniques, the analog and digital approaches, for producing symbol displays on a television screen in response to object data, and they are doing essentially the same thing."[54]

52 See *In re Donaldson Company, Inc.*, 16 F.3d 1189 (Fed. Cir. 1994).
53 216 U.S.P.Q. 28 (N.D. Ill. 1982).
54 *Id.* at 49.

Similarly, in *Magnavox Co., Inc. v. Chicago Dynamic Industries,*[55] infringement of the above patent by digital circuitry was found. The court observed the following:

> The use of digital instead of analog circuitry, it seems to me, is a difference which is not material. I regard analog and digital circuitry as two means which are interchangeable largely, which are equivalent, and which are, therefore, essentially the same means for achieving substantially the same results in substantially the same way.... If one were to say that a mere change from analog to digital circuitry were to be a sufficient change to deprive an analog patent of protection, then it seems to me that every electronic invention would be fair game for anyone who simply used the reverse method of circuitry to achieve the same result.[56]

Decca Ltd. v. United States[57] found infringement of U.S. Patent No. 2,844,816, involving analog circuitry by a programmed computer and held that "infringement is not avoided by showing that digital instead of analog techniques are used..."[58]

In *Hughes Aircraft Co. v. United States,*[59] the Federal Circuit held that a spacecraft, with its ISA position retained on-board, is equivalent to the patentee's claimed satellite, with its ISA position sent to the ground, because they both perform substantially the same function. The claimed invention used a ground-based analog controller for synchronizing the force applied to the satellite with the satellite's spin cycle while the infringing spacecraft used an on-board computer.

In *Lockheed Aircraft Corp. v. United States,*[60] the defendant argued that the structural differences between the claimed invention and the accused digital device showed that the two devices were not equivalent.[61] The U.S. Court of Claims found that the differences were within the scope of the claims of the patent and held that the accused device infringed the plaintiff's patent.

In *Arshal v. United States,*[62] the trial court held that whether the accused apparatus was digital or analog was irrelevant to the question of infringement where the apparatus was intentionally designed to simulate integration as called for by the claim. The U.S. Court of Claims

55 201 U.S.P.Q. 25 (N.D. Ill. 1977).

56 *Id.* at 29.

57 554 F.2d 1070 (1976), *aff'd in part, modified in part, and rev'd in part,* 640 F.2d 1156 (Ct. Cl. 1980), *cert. denied,* 454 U.S. 819 (1981).

58 citing *National Dairy Products Corp. v. Swiss Colony Inc.,* 364 F. Supp. 134, 152 (W.D. Wis. 1972).

59 717 F.2d 1351 (Fed. Cir. 1983).

60 553 F.2d 69 (Ct. Cl. 1977).

61 *Id.* at p. 80.

62 202 U.S.P.Q. 749 (1979), *modified,* 621 F.2d 421 (Ct. Cl. 1980).

modified the trial judge's opinion to delete all references to the infringement question because it held the patent invalid under 35 U.S.C. § 101.

Also, in *Dynamics Corp. of America v. United States*,[63] in the accounting phase of the trial, the U.S. Claims Court held that a hybrid computer (which substituted a computer-calculated voltage for loop-generated voltage) should also be considered an infringing system. See also, the unreported infringement decision in *IVAC Corporation v. Terumo Corp.*,[64] which found infringement of U.S. Patent Nos. 3,702,076 and 3,877,307, under the doctrine of equivalents based on digital/analog equivalence.

Note the other edge of the sword in *James Constant v. Advanced Micro-Devices, Inc.*[65] The invalidity of U.S. Patent No. 3,950,635, which implemented a shift register function with RAM memory was sustained by the Federal Circuit. Additionally, U.S. Patent No. 4,438,491, which covered an integrated circuit having a "transmitting and receiving means," was held anticipated or obvious in view of a prior publication of a computer program.

If the law were changed so that programmed computer embodiments were not within the ambit of patent protection, the outcome of the above-mentioned cases would be the opposite. Also, depending on the complexity of the invention, it would be relatively easy to use the disclosure in a hardware patent to make a programmed computer equivalent, and thereby evade the patent. This, in turn, would undermine our entire precedent relating to patented circuitry, as noted in *Magnavox*.

The Alappat application presented the critical issue directly to the Federal Circuit; either patents cover both programmed and hardwired embodiments or there is a gaping hole in patent protection for electrical inventions. *Alappat* clearly establishes that inventors are entitled under 35 U.S.C. § 112 (6) to claim their inventions in functional terms broad enough to cover a programmed computer and that, so defined, the claimed invention is statutory under 35 U.S.C. § 101.

3. Denying Equivalency Leads to Inconsistencies and Scientifically Untenable Decisions

Inconsistency has been a problem particularly in Europe where computer program-related inventions are usually viewed as unpatent-

63 5 Cl. Ct. 591 (1984).
64 Case No. 87-0413-B(M) (S.D. Cal. 1989).
65 848 F.2d 1560 (Fed. Cir. 1988).

able or less patentable than they have been in the United States.[66] But because programmed computers are simply equivalent to hardwired signal processes, there is no scientific basis for the distinction, which leads to scientifically (and ultimately societally) indefensible decisions.

Consider the following decision and patent in the United Kingdom: *International Business Machines Corporation's Application*[67] and the U.K. Patent 1,352,7,42 which issues therefrom. The invention is a data processing system used to price commodities. The invention disclosed in the specification is directed to a hardwired embodiment, but it is clear that a programmed general purpose computer could easily do the equivalent, which led to the court's decision. The court decided that the invention involved a "special purpose computer" (which is patentable) rather than a programmed "standard computer" (which is not patentable). From a scientific standpoint, this debate is silly—clearly the invention can be practiced either way. Having a patent cover one embodiment without the other is ridiculous.[68]

II. Conclusion

The view of *In re Alappat* that "programming creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed....'" is the key to understanding how patentable subject matter requirements apply to computer program-related inventions. It is simply an application of the plain meaning of 35 U.S.C. § 101 to the facts. The Federal Circuit majority is correct in its approach.

66 Note that the Patent Corporation Treaty limits patenting inventions drawn to computer programs.

67 Patents Appeal Tribunal 1980 F.S.R. 564 (Pat. App. Trib. 1978).

68 This situation undermines (if not negates) the value of patents on hardware and vitiates the intended purposes of patent law. Why disclose how to make and use the invention if a competitor can easily design around it? Absent patent protection, the hardware inventor may be better off not disclosing anything rather than arming competitors with a complete disclosure. Protecting only pure hardware embodiments makes patents less valuable and less attractive, thereby stifling America's creative genius.